Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A method for modifying a graph-based ~~representing~~ representation of an executable computer application, the graph-based representation including a graph having vertices representing components and links between components indicating flows of data between such components, the graph further having components with parameters, including:

~~(a)~~ programmatically retrieving [[a]] definitions of runtime parameters for the graph at runtime execution of the graph, the runtime parameters ~~having a value~~ being defined as determinable at runtime execution of the graph;

~~(b)~~ determining whether ~~the~~ a value for each of the runtime parameters is to be provided by user input; ~~or~~

determining whether a value for each of the runtime parameters is to be externally supplied programmatically;

~~(c)~~ displaying a prompt to a user for receiving user input for every runtime parameter so determined to be provided by user input;

~~(d)~~ retrieving any externally supplied value for every runtime parameter determined to be externally supplied programmatically;

~~(e)~~ determining ~~a final~~ parameter values for the runtime parameters based on ~~one of~~ the user input to such prompt or such externally supplied value or a default value;

~~(f)~~ modifying the graph-based representation of the application ~~represented by the graph~~ using the ~~final~~ determined parameter values ~~as the value~~ for the runtime parameters; and

~~(g)~~ executing the application represented by the modified graph-based representation.

2. (Canceled)

3. (Original): The method of claim 1, further including providing an interface which permits designating a parameter of a graph component as a runtime parameter.

4. (Currently amended): The method of claim 1, wherein determining the ~~final~~ parameter values includes evaluating an expression.

5. (Original): The method of claim 4, wherein the expression computes metadata.

6-7. (Canceled)

8. (Previously presented): The method of claim 1, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

9. (Previously presented): A method for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the method including:

(a) determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having an associated condition and an associated condition-interpretation;

(b) evaluating the associated condition for every such conditional component;

(c) modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by removing such conditional component and all connected flows to such conditional component from the graph before execution of the graph, based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

(d) executing the application represented by the modified graph.

10. (Canceled)

11. (Previously presented): The method of claim 9, further including removing each component and flows connected to such components that depend on the presence of the removed conditional component.

12. (Previously presented): A method for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the method including:

(a) determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having a an associated condition and an associated condition-interpretation;

(b) evaluating the associated condition for every such conditional component;

(c) modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by replacing such conditional component with a flow before execution of the graph based on an evaluation of the associated condition and the corresponding condition-interpretation for such conditional component; and

(d) executing the application represented by the modified graph.

13. (Previously presented): The method of claims 9 or 12, further including providing an interface which permits designating a condition and a condition-interpretation for a graph component.

14. (Currently amended): A system for modifying a graph-based ~~representing~~ representation of an executable computer application, the graph-based representation including a graph having vertices representing components and links between components indicating flows

of data between such components, the graph further having components with parameters, including:

~~(a)~~ means for programmatically retrieving [[a]] definitions of runtime parameters for the graph at runtime execution of the graph, the runtime parameters ~~having a value~~ being defined as determinable at runtime execution of the graph;

~~(b)~~ means for determining whether ~~the~~ a value for each of the runtime parameters is to be provided by user input; ~~or~~

means for determining whether a value for each of the runtime parameters is to be externally supplied programmatically;

~~(c)~~ means for displaying a prompt to a user for receiving user input for every runtime parameter so determined to be provided by user input;

~~(d)~~ means for retrieving any externally supplied value for every runtime parameter determined to be externally supplied programmatically;

~~(e)~~ means for determining ~~a final~~ parameter values for the runtime parameters based on ~~one of~~ the user input to such prompt or such externally supplied value or a default value;

~~(f)~~ means for modifying the graph-based representation of the application ~~represented by the graph~~ using the ~~final~~ determined parameter values ~~as the value~~ for the runtime parameters; and

~~(g)~~ means for executing the application represented by the modified graph-based representation.

15. (Canceled)

16. (Original): The system of claim 14, further including an interface which permits designating a parameter of a graph component as a runtime parameter.

17. (Currently amended): The system of claim 14, wherein the means for determining the ~~final~~ parameter values includes means for evaluating an expression.

18. (Original): The system of claim 17, wherein the expression computes metadata.

19-20. (Canceled)

21. (Previously presented): The system of claim 14, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

22. (Previously presented): A system for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the system including:

(a) means for determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having an associated condition and an associated condition-interpretation;

(b) means for evaluating the associated condition for every such conditional component;

(c) means for modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by removing such conditional component and all connected flows to such conditional component from the graph before execution of the graph, based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

(d) means for executing the application represented by the modified graph.

23. (Canceled)

24. (Previously presented): The system of claim 22, further including means for removing each component and flows connected to such components that depend on the presence of the removed conditional component.

25. (Previously presented): A system for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the system including:

(a) means for determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having a an associated condition and an associated condition-interpretation;

(b) means for evaluating the associated condition for every such conditional component;

(c) means for modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by replacing such conditional component with a flow before execution of the graph based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

(d) means for executing the application represented by the modified graph.

26. (Previously presented): The system of claims 22 or 25, further including an interface which permits designating a condition and a condition-interpretation for a graph component.

27. (Currently Amended): A computer program, stored on a computer-readable medium, for modifying a graph-based ~~representing~~ representation of an executable computer application, the graph-based representation including a graph having vertices representing components and links between components indicating flows of data between such components, the graph further having components with parameters, the computer program comprising instructions for causing a computer to:

~~(a)~~ programmatically retrieve [[a]] definitions of runtime parameters for the graph at runtime execution of the graph, the runtime parameters ~~having a value~~ being defined as determinable at runtime execution of the graph;

~~(b)~~ determine whether ~~the~~ a value for each of the runtime parameters is to be provided by user input; ~~or~~

determine whether a value for each of the runtime parameters is to be externally supplied programmatically;

~~(c)~~ display a prompt to a user for receiving user input for every runtime parameter so determined to be provided by user input;

~~(d)~~ retrieve any externally supplied value for every runtime parameter determined to be externally supplied programmatically;

~~(e)~~ determine ~~a final~~ parameter values for the runtime parameters based on ~~one of~~ the user input to such prompt or such externally supplied value or a default value;

~~(f)~~ modify the graph-based representation of the application ~~represented by the graph~~ using the ~~final~~ determined parameter values ~~as the value~~ for the runtime parameters; and

~~(g)~~ execute the application represented by the modified graph-based representation.

28. (Canceled)

29. (Original): The computer program of claim 27, further including instructions for causing the computer to provide an interface which permits designating a parameter of a graph component as a runtime parameter.

30. (Currently amended): The computer program of claim 27, wherein the instructions for causing the computer to determine the ~~final~~ parameter values include instructions for causing the computer to evaluating an expression.

31. (Original): The computer program of claim 30, wherein the expression computes metadata.

32-33. (Canceled)

34. (Previously presented): The computer program of claim 27, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

35. (Previously presented): A computer program, stored on a computer-readable medium, for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the computer program comprising instructions for causing a computer to:

(a) determine at runtime execution of the graph whether any component of the graph is defined as being a conditional component having an associated condition and an associated condition-interpretation;

(b) evaluate the associated condition for every such conditional component;

(c) modify the graph at runtime execution of the graph in accordance with such evaluation and the associated corresponding condition-interpretation of at least one such conditional component by removing such conditional component and all connected flows to such conditional component from the graph before execution of the graph, based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

(d) execute the application represented by the modified graph.

36. (Canceled)

37. (Previously presented): The computer program of claim 35, further including instructions for causing the computer to remove each component and flows connected to such components that depend on the presence of the conditional component.

38. (Previously presented): A computer program, stored on a computer-readable medium, for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the computer program comprising instructions for causing a computer to:

(a) determine at runtime execution of the graph whether any component of the graph is defined as being a conditional component having a an associated condition and an associated condition-interpretation;

(b) evaluate the associated condition for every such conditional component;

(c) modify the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by replacing the conditional component with a flow before execution of the graph based on an evaluation of the condition and the corresponding condition-interpretation for such conditional component; and

(d) execute the application represented by the modified graph.

39. (Previously presented): The computer program of claims 35 or 38, further including instructions for causing the computer to provide an interface which permits designating a condition and a condition-interpretation for a graph component.